TELKOMNIKA, Vol.17, No.5, October 2019, pp.2355~2369
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v17i5.12929

Improving compliance with bluetooth device detection

Martin Davies[1], Eoghan Furey*[2], Kevin Curran[3]
[1,2]School of Computing, Letterkenny Institute of Technology, Co. Donegal, Ireland
[3]School of Computing, Engineering and Intelligent Systems, Faculty of Computing, Engineering and Built Environment, Ulster University, Northern, Ireland
*Corresponding author, e-mail: eoghan.furey@lyit.ie[2], kj.curran@ulster.ac.uk[3]

Abstract

The number of devices containing Bluetooth chipsets is continuing to rise and there exists a need to stem the tidal wave of vulnerabilities brought by the Bring Your Own Device (BYOD) and Internet of Things (IoT) phenomena. With Bluetooth enabled but discovery mode turned off, auditing for Bluetooth devices, or creating an accurate Bluetooth device hardware log is limited. The software tools and hardware devices to monitor WiFi networking signals have long been a part of the security auditor's arsenal, but similar tools for Bluetooth are bespoke, expensive, and not adopted by most security pentesters. However, this has changed with the introduction of the Ubertooth One, a low-cost and open-source platform for monitoring Bluetooth Classic signals. Using a combination of the Ubertooth One, and other high-power Bluetooth devices, an auditor should now be able to actively scan for rogue devices that may otherwise have been missed. This research examines various hardware combinations that can be used to achieve this functionality, and the possible implications from a compliance point of view, with a focus on the standards used by the Payment Card Industry Data Security Standard (PCI-DSS), and the guidelines offered by the National Institute of Standards and Technology (NIST). We compare the results of scanning with traditional Bluetooth devices as opposed to an Ubertooth/Bluetooth combination. We show how the ability to monitor a larger portion of Bluetooth traffic can highlight serious implications in the compliance landscape of many organisations and companies. We demonstrate that identifying non-discoverable devices with Bluetooth enabled is a crucial element in holistic security monitoring of threats.

Keywords: *bluetooth hacking, hacking, network security, wireless security, wireless sniffing*

1. Introduction

Bluetooth technology has become all-pervasive, with attach rates close to one hundred percent for mobiles and laptops [1]. Even though it is pervasive, there are misconceptions around range and exposure and many organisations overlook potential threats. These threats should be taken seriously, and evaluated as part of an overall wireless security plan. Bluetooth is a stable and well documented technology dating back to 1994 when Ericsson came up with an idea to use a wireless connection to connect items such as an earphone and a cordless headset and the mobile phone. Later in 1998, five companies (Ericsson, Nokia, IBM, Toshiba and Intel) formed the Bluetooth Special Interest Group [2]. Bluetooth was originally intended to be a short-range wireless technology, which primary purpose was to replace wires and cables. It has found widespread uses especially with wireless keyboards, mice, headsets and hands-free kits.

The volume of Bluetooth devices is growing however, and due to the nature of the technology, they have been difficult to monitor. Until recently, Bluetooth has relied upon security by obscurity. Bluetooth technology is making its way into all kinds of devices, and is attractive due to its low cost and minimal resource requirements. Devices such as Bluetooth Access Points (AP) are available that provide similar connectivity and range as their 802.11 counterparts but escape analysis mechanisms since Bluetooth operates using Frequency Hopping Spread Spectrum (FHSS) instead of traditional 802.11 transmission mechanisms [3]. With the introduction of the Ubertooth One, this monitoring and analysis can now take place in a cost-effective and efficient manner. With the Ubertooth's ability to capture the lower 4 bytes of the Bluetooth Device Address (BD_ADDR), a standard Bluetooth dongle could be used to actively enumerate identified Bluetooth devices, the combination of the two Universal Serial Bus (USB) devices would provide the needed information to quickly and

Received October 30, 2018; Revised March 23, 2019; Accepted April 25, 2019

accurately characterise devices in the area [4]. This device capture and categorisation is very important. A rogue AP is any device that adds an unauthorised (and therefore unmanaged and unsecured) WLAN to the organisation's network [5].

We test the above assertions in this work using various USB Bluetooth devices in combination with the Ubertooth One. These devices range from the built-in low-end device, through to a high-power industrial device, exploring the important capability of creating a usable hardware inventory. As an ad-hoc technology, Bluetooth devices are often utilised within organisations, outside of the control of IT management. Few organisations recognise the threat of Bluetooth technology, often due to misconceptions in the technology, and the threats of use. Now armed with a hardware inventory appended with previously unavailable Bluetooth devices, it should be possible for the audit professional to identify known APs and Bluetooth stations from rogue devices. By doing this, this research aims to capture and identify non-discoverable devices, and reveal that a sizable proportion of additional devices, with Bluetooth enabled, can now be captured, highlighting a gap in existing Bluetooth auditing practices.

2. Bluetooth

The Institute of Electrical and Electronics Engineers (IEEE) created a family of standards called IEEE 802 that deal with Local Area Networks (LAN). Two of the most relevant working groups of this standard are 802.11, which deals with wireless LANs, and 802.15 which specifies Wireless Personal Area Networks (WPAN). Bluetooth operates within the 802.15 standard, and uses the same frequency range (2.4 GHz) as 802.11 (WiFi). Bluetooth uses the licence free Industrial, Scientific and Medical (ISM) frequency band for its radio signals and enables communications between devices up to a maximum distance of about 100 m, although it is normally used for shorter distances [2]. The Bluetooth channels are spaced 1 MHz apart, beginning at 2402 MHz and ending at 2480 MHz. This arrangement of 79 individual Bluetooth channels gives a guard band of 2 MHz at the bottom and 3.5 MHz at the top which is presented in Table 1. It should be noted that the 2472 MHz and 2480 MHz bands are outside the standard operating frequencies for WiFi (in the US).

Bluetooth devices transmit in the 2.4 GHz band using Frequency-Hopping Spread Spectrum (FHSS). Frequency hopping is a novel way to avoid busy channels used by other devices, WiFi or microwave ovens for example. A Bluetooth transmission remains only on a given frequency for a brief time, unlike WiFi for example, and if any interference is present the data will be re-sent later when the signal has changed to a different channel which is likely to be clear of other interfering signals. The standard uses a hopping rate of 1600 hops per second, and the system hops over all the available frequencies using a pre-determined pseudo-random hop sequence based upon the Bluetooth address of the master node in the network. Bluetooth hardware is rated in classes, which regulates the transmission output power, and therefore range of the devices. The common power classes are displayed in Table 2. Note that one device, the Aircable Host XR has a far more powerful transmitter than the other class 1 device used for this work. In general, Bluetooth is a short-range technology designed to communicate up to distances of 10 metres. However, longer ranges are possible that cover far greater distances (up to 1 kilometre in perfect conditions using the more powerful 100 mW class 1 devices.

Table 1. Bluetooth Channels and Their Respective MHz Band

Chl	MHz	Chl	MHz	Chl	MHz	Chl	MHz	Chl	MHz	Chl	MHz	Chl	MHz	Chl	MHz
1	2402	11	2412	21	2422	31	2432	41	2442	51	2452	61	2462	71	2472
2	2403	12	2413	22	2423	32	2433	42	2443	52	2453	62	2463	72	2473
3	2404	13	2414	23	2424	33	2434	43	2444	53	2454	63	2464	73	2474
4	2405	14	2415	24	2425	34	2435	44	2445	54	2455	64	2465	74	2475
5	2406	15	2416	25	2426	35	2436	45	2446	55	2456	65	2466	75	2476
6	2407	16	2417	26	2427	36	2437	46	2447	56	2457	66	2467	76	2477
7	2408	17	2418	27	2428	37	2438	47	2448	57	2458	67	2468	77	2478
8	2409	18	2419	28	2429	38	2439	48	2449	58	2459	68	2469	78	2479
9	2410	19	2420	29	2430	39	2440	49	2450	59	2460	69	2470	79	2480
10	2411	20	2421	30	2431	40	2441	50	2451	60	2461	70	2471		

Table 2. Bluetooth Classes, Power and Typical Ranges

Class	Maximum Permitted Power (mW)	(dBm)	Typical Range
Class 1	200 mW (Aircable Host XR)		>100 metres
Class 1	100 mW	20 dBm	~100 metres
Class 2	2.5 mW	4 dBm	~10 metres
Class 3	1 mW	0 dBm	~1 metre

Class 1 devices can increase or decrease their transmission power to the appropriate level based on the Received Strength Signal Indictor (RSSI) reading. This has implications when attempting to physically track a device. Class 2 and 3 devices do not have this capability, as they seek to conserve power and focus on shorter communication distances. In addition to the range of a device, the data transfer rate depends on which version of Bluetooth is supported on the particular device. Table 3 shows the different possible data transfer rates. Table 3 is particularly important, as some of the hardware described later does have limitations in the Bluetooth versions they support.

Table 3. Bluetooth Versions Data Transfer Comparison

Bluetooth Version	Data rate	High Data Rate Traffic	Release Year
1.2	1 Mbit/s	721 Kbit/s	2003
2.0 +EDR	3 Mbit/s	>80 Kbit/s	2007
3.0 HS	24 Mbit/s	802.11 link	2009
4.0	24 Mbit/s	802.11 link	2013

The data rates in Table 3 are not particularly high, the higher data rates cited in the table are only achieved by utilising WiFi, specifically the IEEE 802.11g physical layer (not Bluetooth). Bluetooth packets start with a code that is based on the Lower Address Part (LAP) of a particular Bluetooth Device Address (BD_ADDR). The BD_ADDR is a 48-bit MAC address, just like the MAC address of an Ethernet device. The LAP consists of the lower 24 bits of the BD_ADDR and is the only part of the address that is transmitted with every packet [6]. The BD_ADDR structure is described in more detail in Table 4. For example, with a Bluetooth device address of 11:22:33:44:55:66, you only need to know 00:00:33:44:55:66 to communicate with the device. Since the Upper Address Part (UAP) is only 8 bits long, if you have the LAP, you will quickly be able to interact with the device, as you only need 2^8 (256) at most guesses, before it is found. Finding the LAP is key in terms of security.

Table 4. Bluetooth Device Address Structure

	NAP Non-significant Address Part	UAP Upper Address Part	LAP Lower Address Part
Bits	16 bits	8 bits	24 bits
Sample	00:00	33	44:55:66
		Error check based on UAP CRC also based on UAP	Access Code is derived from LAP
			Manufacturer ensures this part is unique

According to [6], this process can be speeded up by prioritising common UAPs, possible due to the UAP being part of the Organisationally Unique Identifier (OUI) assigned to a relatively small number of manufacturers. Two types of Bluetooth link that are available and can be set up are Asynchronous Connection-oriented Logical (ACL) and Synchronous Connection Orientated (SCO) communications links [2]. ACL is the more widely used. Three main elements that are included in the higher layer stack or Bluetooth host are Logical Link Control and Adaptation Protocol (L2CAP), Service Discovery Protocol (SDP) and Generic Access Protocol (GAP). L2CAP is used to provide an interface for all the data requests that use the ACL links. The Bluetooth L2CAP affords multiplexing between the higher layer protocols which enables several applications to use the same lower layer links. SDP allows devices to discover which services other Bluetooth devices support, and list what the Bluetooth device supports. Bluetooth GAP describes how Bluetooth devices can discover each other and establish connections. It is

one of the most basic Bluetooth profiles, but is used by every other profile as the foundation for establishing a link. Bluetooth GAP can put the device into three different modes of discovery. General discovery, Limited discovery and Non-discoverable. For Bluetooth devices to converse correctly, Bluetooth Profiles are required. Bluetooth profiles are additional protocols that build upon the basic Bluetooth standard to more clearly define what kind of data a Bluetooth module is transmitting. While Bluetooth specifications explain how the technology works, profiles explain how it is used. Of note in this list, from a security point of view, is item 24 OBject EXchange (OBEX) and item 30 Service Discovery Application Profile (SDAP).

Bluetooth Low Energy (BLE) technology was introduced in 2010, through the Bluetooth v4.0 specification. With its low power consumption and new features, BLE enables new applications that were impractical with Bluetooth Classic technology. BLE is an exciting and rapidly growing area of Bluetooth, providing functionality where low power may be a necessity, for example where a device is battery powered, but needs to be available for months or years. The BLE standard offers several advantages over Bluetooth Classic, including low cost, low peak, average and idle mode power consumption, small in size making them useful for accessories and Human Interface Devices (HID). BLE connections are quite simple, more so than the hop pattern of Bluetooth Basic Rate (BR) [7]. Table 5 focusses on these key differences between Bluetooth BR and Enhanced Data Rate (EDR) versus Low Energy. Two items of note from this table (highlighted in red) are the reduced number of channels, and the low maximum output power allowed (resulting in reduced ranges).

Table 5. Key Differences Between Bluetooth BR/EDR and BLE [8]

Characteristic	Bluetooth BR/EDR	Bluetooth LE
RF Physical Channels	79 channels, 1 MHz channel spacing	40 channels, 2 MHz channel spacing
Discovery/Connect	Inquiry/Paging	Advertising
Number of Piconet Slaves	7 (active) / 255 (total)	Unlimited
Device Address Privacy	None	Private Device Addressing available
Max Data Rate	1-3 Mbps	1 Mbps via GFSK modulation
Encryption Algorithm	E0/SAFER+	AES-CCM
Typical Range	30 metres	50 metres
Max Output Power	100 mW (20dBm)	10 mW (10 dBm)

2.1. Discoverable Mode

A device is said to be in discoverable mode when it periodically checks whether other devices are looking for them. Due to the sophisticated nature of Bluetooth technology, and specifically FHSS, Bluetooth creates its connections in a complicated manner. These come in the form of Master-Slave connections, these connections remain in place until they are broken, either by a disconnection, or by a poor-quality link that makes communications impossible (i.e. the devices go out of range). Bluetooth devices have two modes: a discovery mode and a pairing mode. Discovery mode determines how the device reacts to inquiries from other devices looking to connect, and it has three actions. The discoverable action has the device respond to all inquiries, limited discoverable restricts that action, and non-discoverable tells the device to ignore all inquiries. It is a security risk to leave a device in discovery mode [9].

Pairing should be controlled and mutual authentication should be practiced. Historically, Bluetooth security recommendations included turning off discoverable mode. The National Institute of Standards and Technology (NIST), highlights that discoverable devices are more prone to potential attack. Bluetooth device owners may be unaware of their device's inherent vulnerabilities [10]. Being able to retrieve a Bluetooth devices' BD_ADDR is all that is required to establish a connection with a remote device. Many devices rely on this secrecy of the BD_ADDR for security. To facilitate this, Bluetooth devices can be configured in discoverable mode, where they answer page request messages from other devices with their BD_ADDR information, and in non-discoverable mode, where they ignore requests for the BD_ADDR. Turning off discoverability does nothing to thwart skilled attacker and can create a false sense of security [11].

2.2. Non-Discoverable Mode

Credit card skimming devices will be configured in non-discoverable mode if hackers wish to evade detection [12]. A device is said to be non-discoverable if it simply ignores (or does

not look for) discovery requests [13]. Many devices are not discoverable by default, so you must enable this feature specifically, usually for a brief period. Keeping a device in non-discoverable mode is a standard security practice, but is not a security fix. Unlike IEEE 802.11, Bluetooth does not transmit the full BD_ADDR, which makes it possible to capture the last three bytes of the BD_ADDR (LAP). Once these three bytes are known, a user can send connection request messages to every common BD_ADDR prefix, or OUI until, the full BD_ADDR found. In other words, the most important passive Bluetooth monitoring function is simply capturing the LAP from each packet transmitted on a channel. LAP sniffing allows you to identify Bluetooth devices operating in your vicinity. A hardware limitation, until recently was this inadequacy of Bluetooth devices. The Azio Bluetooth adaptor is an active device and can only discover devices that have discovery mode enabled. This device has little value working in proximity to devices in non-discoverable mode. However, this specific drawback/constraint is addressed by the Ubertooth One.

2.3. Bluetooth Security Issues

Bluetooth technology has been integrated into many types of business and consumer devices, including cell phones. Laptops, automobiles, medical devices, printers, keyboards, mice and headsets [8]. This can lead to problems, because security setting will be different on each device, making it difficult to follow generic security advice. Due to the large volume of devices, and a plethora of device types, Bluetooth security is a big issue. Many devices are vulnerable to an excess of attacks included denial of service (DOS), main-in-the-middle (MITM) attacks, eavesdropping, etc. Some of the more common Bluetooth attacks/vulnerabilities, are presented in Table 6.

Table 6. Common Bluetooth Attacks/Vulnerabilities

Attack Name	Description
Bluebug attack	An attacker can use the AT commands on a victim's cell phone to initiate calls, send SMS messages [9]. This form of Bluetooth security issue allows hackers to remotely access a phone and use its features. This may include placing calls and sending text messages.
Bluejacking	Allows an anonymous message to be displayed on the victim's device [9]. Often, the Bluejacker is trying to send someone else their business card, which will be added to the victim's contact list in their address book [14].
Bluescarfing	Bluescarfing is the actual theft of data from a mobile device [8].
Bluesnarfing	Bluesnarfing is the unauthorised access from a wireless device through a Bluetooth connection. This allows access to a calendar, contact list, e-mails, and text messages, and on some phones, users can copy pictures and private videos [14].
Bluesmacking	Bluesmacking is simply a denial-of-service attack against a device [8].
Bluesniffing	Bluesniffing is exactly what it sounds like [8].
Buffer overflow	Buffer overflow: An attacker can remotely exploit bugs in the software on Bluetooth-enabled devices [9].
Car Whispering	Car Whispering: This involves the use of software that allows hackers to send and receive audio to and from a Bluetooth enabled car stereo system [2].

This all leads to difficulties for the individual with responsibility for the Bluetooth region of the attack landscape. Without a means to monitor active Bluetooth devices, nor having the capability to passively sniff those device's traffic, it is difficult to determine if any attacks took place, eavesdropping traffic for example.

2.4. Intrusion Detection

An Intrusion Detection System (IDS) is a system used to determine whether unauthorised access (intrusions) are occurring on a network. Once identified, mitigating steps can be initiated, perhaps using an Intrusion Prevention System (IPS). Intrusion detection is the process of monitoring the events occurring in a computer system or network and analysing them for signs of possible incidents which are violations or imminent threats of violation of security policies [1]. An IPS is a system that has all the capabilities of an IDS but can also attempt to stop possible incidents. IDS/IPS are well established in Wi-Fi (802.11) but are limited in their Bluetooth (802.15) support, because of the volume of such devices and the nature of the technology. There are three main categories of IDS, Network-based Intrusion Detection

System (NIDS), Host-based Intrusion Detection System (HIDS) and Distributed Intrusion Detection System (DIDS).

In all cases, the DIDS's defining feature requires that the distributed sensors report to a central management station [15]. While no central management is described, it is the feasibility of the sensor itself that is being tested. If a successful sensor combination is found, it can be added to existing security toolsets. An IDS is a critical component in a defence-in-depth information security strategy [15]. Defence in depth is the method of protecting information resources with a series of overlapping defensive mechanisms. The idea being if one defence fails, others will thwart an attack. These systems are difficult to implement, with false positives being very common in initial stages while issues are being ironed out. Logging devices is very important. In combination with an up to date hardware inventory a wireless IDS/IPS should be able to observe all APs and clients, on all operational channels, and classify each device as authorised, unauthorised/rogue or neighbouring. A SYSLOG type system would be useful for this purpose.

3. Compliance and Guidelines

We investigate how the use of wireless technologies, and Bluetooth, can affect the overall compliance landscape of an organisation, with particular emphasis on the standards used by the Payment Card Industry Data Security Standard (PCI-DSS), who require that organisations regularly assess their networks for these rogue AP threats, and many vendors have implemented products designed to address this threat.

3.1. PCI-DSS

Created by four credit card companies in 2004 Visa, MasterCard, American Express and Discover; the PCI-DSS provides a minimum set of requirements created by the PCI Security Standards Council. The purpose of these standards is to help protect credit card data. The full specifications of PCI-DSS are available at the PCI Standards Security Council website, and are summarised in Table 7, which highlights the six main objectives and twelve requirements. It should be noted that these are minimum requirements. Being PCI compliant does no meant that the data is completely safe from attack.

Table 7. PCI-DSS Objectives and Requirements

PCI–DSS Objectives	PCI-DSS Requirements
1. Build and Maintain a Secure Network and Systems	1. Install and maintain a firewall configuration to protect cardholder data 2. Do not use vendor-supplied defaults for system passwords and other security parameters
2. Protect Cardholder Data	3. Protect stored cardholder data 4. Encrypt transmission of cardholder data across open, public networks
3. Maintain a Vulnerability Management Program	5. Protect all systems against malware and regularly update anti-virus software or programs 6. Develop and maintain secure systems and applications
4. Implement Strong Access Control Measures	7. Restrict access to cardholder data by business need to know 8. Identify and authenticate access to system components 9. Restrict physical access to cardholder data
5. Regularly Monitor and Test Networks	10. Track and monitor all access to network resources and cardholder data 11. Regularly test security systems and processes
6. Maintain an Information Security Policy	12. Maintain a policy that addresses information security for all personnel

Several of these requirements are affected by the inherent weaknesses of Bluetooth, and they do require a lot of work to implement. [16] advises maintaining the physical security of wireless data and having a person at each physical location responsible for checking if equipment has been tampered with or compromised in any way. This person must manually assess (utilising vendor guidance) the security of the access points, wireless controllers, and any other physical pieces of the organisation's WLAN. The PCI-DSS Security Standards Council recommends periodic detection and identification of unknown and potentially dangerous rogue wireless devices, as well as documented response procedures in the event unauthorised wireless devices are detected. Which of course is particularly difficult for Bluetooth. In order for effective detection to take place, it is vital that an updated hardware inventory, including

Bluetooth devices (BD_ADDR and friendly device name information), be constantly updated and maintained. This is important so legitimate devices can be distinguished from illegitimate devices. Besides actively scanning the ISM band, physical and manual inspections of APs, hardware and networking devices is also important, as it may indicate whether unauthorised devices are connected or not. This physical inspection, will not however, tell an auditor if devices had been connected in the past, and subsequently removed.

When a rogue device is discovered, it then needs to be logged, and/or disabled. A verification scan could also be run. Of relevance to Bluetooth are requirements 11 and 12. The standards indicate that rogue threats need to be immediately resolved, with the environment rescanned as soon as possible. Protecting the data in its own environment is of paramount importance to the PCI-DSS standards, where it is categorised as the Cardholder Data Environment (CDE), and is comprised of people, processes, and technology that store, process, or transmit cardholder data or sensitive authentication data. The PCI-DSS specification is specific in its definition of CDE, and how it comes into scope, or not, for a wireless (Bluetooth) network. To be out of scope from a PCI audit, it must be completely isolated from the CDE, with no possibility of traffic between the two environments. The PCI-DSS wireless Special Interest Group offer some specific Bluetooth recommendations summarised in Table 8. However, in relation to item 2, as previously stated, turning a device to undiscoverable is now no longer an effective defence, due to the passive scanning ability of the Ubertooth One. We believe this recommendation could be updated to state explicitly that a better defence would be to turn Bluetooth off, unless required, and then only turned on when needed.

Table 8. PCI-DSS Bluetooth Recommendations

	Bluetooth Configuration Recommendations
1.	Choose PIN codes that are sufficiently random and long. Avoid static and weak PINs, such as all zeroes.
2.	Bluetooth devices should be configured by default as, and remain, undiscoverable except as needed for pairing.
3.	Ensure that link keys are based on combination keys rather than unit keys. Do not use unit keys.
4.	For v2.1 device using Secure Simple Pairing, do not use the Just Worksll model.
5.	Perform service and profile lockdown of device Bluetooth stacks. Do not allow the use of multiple profiles in the unit.
6.	In the event a Bluetooth device is lost or stolen, immediately unpair the missing device from all other Bluetooth devices with which it was previously paired.

3.2. NIST Special Publications

Besides the direction given by the Payment Card Industry, the National Institute of Standards and Technology (NIST) also provide several Special Publications (SP) in the 800 series, which are of particular interest to the computer security community. The three most relevant publications for this research are described as:

a. NIST SP 800-121 (Revision 1) Guide to Bluetooth Security which supersedes NIST SP 800-121 Guide to Bluetooth Security [17] offers a thorough Bluetooth Mitigation Checklist of 33 items. Nevertheless, with regards to the checklist itself, for item 6, maintaining a complete checklist should be a recommended practice. This checklist would enable an auditor to identify rogue devices, while helping to trace the origin of these rogue devices. For item 33, designating an individual to track the progress of security Bluetooth products, should also be a recommended practice as different threats are being created, and different vulnerabilities are being exploited. As these issues are identified, they can be tracked and addressed by an individual with the right subject matter expertise. Item 16 in the list Bluetooth devices should be configured by default as undiscoverable and remain undiscoverable except as needed for pairing, is particularly relevant to this research, as a passive sniffing device like the Ubertooth One does not care whether this setting is turned on or not.

b. NIST SP 800-124 (Revision 1) Guidelines for Managing the Security of Mobile Devices in the Enterprise which supersedes NIST SP 800-124 Guidelines on Cell Phone and PDA Security is high level and offers advice on policies to manage mobile devices in the enterprise. Some of the more general advice offered around Bluetooth include limiting user access and application access to hardware devices, including Bluetooth, while actively managing wireless interfaces (Bluetooth and WiFi for example).

c. NIST SP 800-94 Guide to Intrusion Detections and Prevention Systems (IDPS) offers guidance on wireless Intrusion Detection outlining how it is most commonly deployed within range of an organisation's wireless network to monitor it, but also can be deployed to locations where unauthorised wireless networking could be occurring. This document unfortunately has no literature covering Bluetooth specifically but does provide important advice for WiFi that may be transferrable, such as recommendations on sensor locations, recommending that wireless sensors actively monitor Radio Frequency (RF) ranges used by the organisation. Also offered is a valuable guide to data fields that should be logged by such devices, as depicted in Table 9, which would be helpful for developing SYSLOG type functionality.

Table 9. Data Fields for Wireless IDS Logging [18]

Data fields for Wireless IDS logging
Timestamp (usually date and time)
Event or Alert type
Priority or severity rating
Source MAC address (the vendor is often identified from the address)
Channel number
ID of the sensor that observed the event
Prevention action performed (if any)

Bluetooth and WiFi have a lot in common, including the sharing of the same ISM range. [19] outlined three distinct steps required, common to both technologies, used to collect all the pertinent data for each device. Step one is to command the hardware to scan all available channels for discoverable devices and return their MAC addresses. Step two was to list the MAC addresses gathered, each device could then be queried to determine the device's human friendly name. Finally, with the device's MAC and name recorded, the system can then use Service Discovery Protocol to find out the high-level services the target device offers.

One of the disadvantages of the original version of Bluetooth was that the data rate was not sufficiently high, when compared to other wireless technologies such as 802.11 [2]. In 2004, an updated version of Bluetooth, known as Bluetooth 2 was ratified which delivered enhanced speeds (EDR) increasing the maximum data rate to 3 Mbps, a significant increase on what was available in the previous Bluetooth specifications. However, it was not until Bluetooth Version 3, until aspects of both technologies merged. Bluetooth 3 enables these much higher speeds by utilising a collocated IEEE 802.11 link, the Bluetooth link being used for the negotiation and establishment of the WiFi connection. Even though Bluetooth was now using 802.11 technology to enjoy higher speeds, it lacked the generic wireless sniffing tools that generally available in the WiFi arena. At present, many WiFi devices have the capability to monitor, and tamper with, wireless networks. Until recently, this capability was not cheaply available with Bluetooth devices [19], if it was available, an auditor could actively monitor the Bluetooth spectrum. However, the Ubertooth One makes it possible for this active monitoring to take place. One thing that sets the Ubertooth apart from other Bluetooth platforms is its capability of not only sending and receiving 2.4 GHz signals, but also operating in monitor mode, monitoring Bluetooth traffic in real-time. This mode has been available in commodity WiFi modules for years and has found myriad uses in research, development and security auditing but no such solution existed for the Bluetooth standard until now.

4. Wireless Analyser Setup

Wireless analysers range from free tools to more expensive commercial scanners, whose purpose is to sniff for wireless devices within the vicinity and identify them. By doing this an auditor can audit a site, and then manually investigate rogue signatures to determine if the device has access to the CDE. In this way devices could be classified as rogue, authorised or a neighbouring device. While this works for WiFi for Bluetooth a new toolset is required.

4.1. BlueZ

BlueZ is a powerful Bluetooth communications stack with extensive APIs that allow a user to fully exploit all local Bluetooth resources. It is open source, freely available, and comes with all major distributions of GNU/Linux [19]. There are three parts to a Bluetooth subsystem on

Linux, the kernel routines, the libbluetooth library, and the six user tools. The user tools are indispensable when configuring or modifying Bluetooth devices on a machine and debugging applications [20]. These commands are described in Table 10.

Table 10**.** Bluetooth Linux Tools Quick Reference [20]

Tool Name	Tool Description
hciconfig	Configure the basic properties of local adapters
hcitool	Detect nearby devices; display information on and adjust low-level connections
sdptool	Search for and browse SDP services. Basic configuration of locally advertised services
hcidump	Low-level debugging of connection setup and data traffic
l1ping	Test L2CAP connection functionality
uuidgen	Generates unique UUID for use with SDP

The command hciconfig is used to configure the basic properties of Bluetooth adapters. As the name suggests, it provides a user-level interface to the (HCI) protocol. When invoked without any arguments, it will display the status of the adapters attached to the local machine [20]. By running the hciconfig command without any options, the connected devices are displayed. The hcitool can be used for Bluetooth discovery and basic enumeration. When scanning, hcitool caches information about devices, reporting the presence of devices that were once observed but may no longer be in range. By default, hcitool shows only BD_ADDR and device name information, but can collect additional details by adding the all parameter. This tool can search and detect nearby Bluetooth devices and show information about low-level Bluetooth connections.

4.2. Ubertooth-Scan

Ubertooth allows low-level Bluetooth data to be captured showing non-discoverable devices in the area. ubertooth-scan uses the LAP recovery feature of ubertooth-rx with an Ubertooth interface, but it also uses the Linux BlueZ Bluetooth interface with a traditional Linux dongle to validate a potential NAP for the identified LAP. ubertooth-scan speeds up NAP recovery while eliminating false-positives [12]. Ubertooth-scan requires both an Ubertooth and a standard Bluetooth device on a host with BlueZ installed. The tool uses the Ubertooth to passively sniff for Bluetooth packets, retrieving the LAP (and eventually) UAP values before handing them to libbluetooth to query the device name. ubertooth-scan was the primary survey tool used. Table 11 shows sample outputs from the three different kinds of scans that ubertooth-scan tool is capable of delivering.

Table 11. Ubertooth-scan Options Explanation

Command	Description	Command Features				
		Initiate Standard Device Scan	hcitool Type Scan	Check for Supporting Features	Chipset Version	Clock Offset
ubertooth-scan	Basic Scan	✓				
ubertooth-scan -s	HCI Type Scan	✓	✓			
ubertooth-scan -x	Extended Scan	✓		✓	✓	✓

The LAP of nearby devices appeared to be quite easy to find in each case and this was clearly seen in all three scan types, as they can find LAPs and their corresponding UAPs. The basic scan appears to ignore discoverable devices, and concentrates only on the non-discoverable ones for LAP capture and UAP enumeration. The HCI type scan does not actually use the BlueZ hcitool, but it does call the same library functions (libbluetooth), and performs the equivalent of running the command: hcitool scan to return their full Bluetooth Device Address and the device friendly name. The devices shown for this part of the scan are discoverable devices. The HCI type scan then continues with capturing non-discoverable LAPs and attempting to discover their corresponding UAPs. The extended scan uses the Ubertooth One to find devices that are transmitting within range, prior to offloading to your Bluetooth device to perform the extended scan on the devices found [21]. 4 bytes of the BD_ADDR

address are required for this to work. This command should produce output similar to running the BlueZ command: hcitool info.

4.3. Ubertooth One

Michael Ossmann presented Ubertooth Zero at Toorcon in 2010. Prior to this, sniffing Bluetooth in a similar fashion to WiFi was both difficult and expensive. Regular Bluetooth dongles just did not have the necessary passive scanning functionality, [22] highlighted that active Bluetooth scanning could be performed using commodity Bluetooth devices, however passive scanning required specialist hardware and software libraries. The Ubertooth Zero was such a specialist hardware and software product. In 2011 at ShmooCon, Ossmann presented the Ubertooth One. Until then, expensive industrial equipment, or specialised software defined radios (SDR) were the only option to sniff Bluetooth packets. The Ubertooth One, was an off-the-shelf product with a power rating comparable to a Class 1 Bluetooth device. Shown in Figure 1 is the reverse side of the printed circuit board (PCB) of the Ubertooth One, the device is Open Source and employs a Joint Test Action Group (JTAG) interface for hardware debugging purposes.

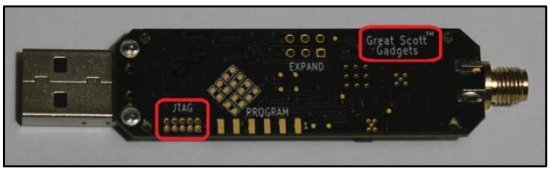

Figure 1. Ubertooth one, from great scott gadgets, note the JTAG pins

Some of the main features of the Ubertooth One are 2.4 GHz transmit and receive, transmit power and receive sensitivity comparable to a Class 1 Bluetooth device, Standard Coretex Debug Connector (10-pin 50-mil JTAG), In-System Programming (ISP) serial connector and an expansion connector: intended for inter-Ubertooth communication or other future uses. Shown in Figure 2 is the operational side of the Ubertooth One PCB. The key components of the device is outlined and highlighted, including the various indicator Light Emitting Diodes (LEDs) that are used.

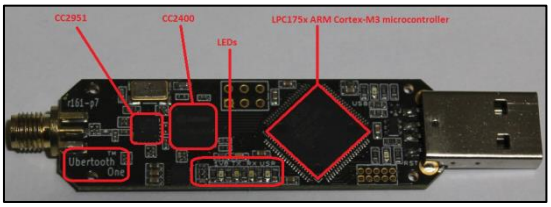

Figure 2. Ubertooth one, showing key components

The Ubertooth One is based on the Texas instruments CC2400 which demodulates raw bits from the air and stream them to a microcontroller. Unfortunately, the Ubertooth hardware is incompatible with Bluetooth Enhanced Data Rate (EDR) data modulations. Even though this EDR limitations exists, fortunately Bluetooth Low Energy does work. Although it was originally built to monitor classic Basic Rate (BR) Bluetooth, it can be repurposed as a BLE sniffer.

4.4. Linksys USBBT100

Two class 1 Linksys USBBT100 USB Bluetooth adaptor were used as adapters with external antennas have a better range out of the box. They are also easier to modify for use with a larger antenna [19]. Most vendors do not design dongles with external antenna connectors. However, with a pigtail/antenna attached, the range of a Class 1 dongle can be extended. Bluetooth devices operate in the 2.4 GHz spectrum, so can use antennas designed for WLAN devices. Technical specifications for the USBBT100 can be found in Table 12. It

should be noted, that while both these devices were from Linksys, the unmodified device used a chipset from Cambridge Silicon Radio (CSR), while the modified device (with a pigtail) used a Broadcom Corporation chipset. Besides the use of different chipsets, the HCI version and LMP version was also different. The table offers a comparison of the main features of both devices. Information was also taken from the hciconfig a command.

Table 12. Linksys USBBT100 Specifications

Name	Linksys Unmodified Device	Linksys Modified Device
Manufacturer	Linksys	Linksys
Name	USBBT100	USBBT100
Power Class	Class 1 (13~17 dBm)	Class 1 (13~17 dBm)
Antenna	1.2 dBi	5 dBi (attached to pigtail)
BD Address	00:0C:41:E2:77:7B	00:13:10:5D:3F:55
HCI Version	1.1	1.2
LMP Version	1.1	1.2
Manufacturer ID	Cambridge Silicon Radio (10)	Broadcom Corporation (15)
Bluetooth Specification	Bluetooth Core Specification 1.1	Bluetooth Core Specification 1.1

Further investigation of these HCI and LMP versions, indicated that both these devices support the Bluetooth Core Specification 1.1 which means the Data Transfer rate is limited to 1 Mbit/s.

4.5. Aircable Host XR

The Aircable Host XR is a USB Bluetooth device that is primarily designed for proximity marketing. It has a 200-mW radio (twice the power of a normal Class 1 device) and a standard RP-SMA antenna connector, which makes it perfect for long range applications [19]. As this device had the most powerful radio transmitter it was used with the larger 9 dBi Antenna. The specifications for this device are summarised in Table 13.

Table 13. Aircable Host XR Specifications

Name	Aircable Host XR
Manufacturer	Aircable
Name	Host XR
Power Class	Class 1 (19.5 dBm)
Antenna	9 dBi
BD Address	00:50:C2:7F:47:80
HCI Version	2.0
LMP Version	2.0
Manufacturer ID	Cambridge Silicon Radio (10)
Bluetooth Specification	Bluetooth Core Specification 1.2

The HCI and LMP versions taken from running the hciconfig command indicated that this device supports the Bluetooth Core Specification 1.2 which means the Data Transfer rate is limited to 1 Mbit/s.

4.6. SENA Parani UD-100

Only a limited number of commercial Bluetooth adaptors are available with external antenna connectors. These are typically intended for industrial type applications. One product is the SENA Parani UD-100 adaptor with a RP-SMA antenna connector. This product also has the advantage of using the CSR chipset. The Parani UD100 from SENA is a high-performance Class 1 Bluetooth adapter that can extend the effective range of Bluetooth up into the hundreds of metres. This Class 1 adapter is much smaller and lighter than other high-performance hardware from companies such as Aircable, which makes it a natural choice for mobile work. A 5 dBi antenna was attached to the Sena, during the testing phase. This proved a discrete and powerful device, a summary of its specifications can be found in Table 14. The HCI and LMP versions taken from running the hciconfig command indicated that this device supports the Bluetooth Core Specification 2.1 + EDR meaning the Data Transfer rate was limited to 3 Mbit/s.

Table 14. SENA Parani UD100 Specifications

Name	SENA Parani UD-100
Manufacturer	SENA
Name	Parani UD100
Power Class	Class 1 (19 dBm) + 6 dBm EDR
Antenna	5 dBi
BD Address	00:01:95:21:C4:95
HCI Version	4.0
LMP Version	4.0
Manufacturer ID	Cambridge Silicon Radio (10)
Bluetooth Specification	Bluetooth Core Specification 2.1 + EDR

5. Evaluation

Each device plugged into 1 central powered hub could be brought back up individually, using hciconfig hciX up, and tested in combination with the Ubertooth One. Performing the testing in this way allowed for all the commands to be run from a Bourne Again Shell (Bash) shell script. It was important to record the hciconfig configuration data at the start of each run, as this can vary each time the machine is rebooted and may be different each time. To help with this task each device was individually identified first, the identification information captured is recorded in Table.

Table 15. Bluetooth Test Devices Individual BD_ADDR Addresses

Device Name	Bluetooth Device Address	Vendor
Device 1: Thinkpad Bluetooth Device	BD_ADDR: 78:DD:08:B2:DE:4C	Hon Hai Precision Ind. Co.,Ltd.
Device 2: Linksys USBBT100	BD_ADDR: 00:0C:41:E2:77:7B	Cisco-Linksys, LLC
Device 3: Linksys USBBT100 (modified)	BD_ADDR: 00:13:10:5D:3F:55	Cisco-Link
Device 4: Parani UD100	BD_ADDR: 00:01:95:21:C4:95	Sena Technologies
Device 5: Aircable Host XR	BD_ADDR: 00:50:C2:7F:47:80	ieee registration authority

The vendor column in the table 15 was discovered using the first 6 characters of the BD_ADDR (NAP and UAP). Besides using the MAC address to discover the manufacturer of the device, it is also possible to discover the chipset maker using the official Bluetooth company identifiers. Several bash scripts were created to identify devices in hidden-mode/ non-discoverable mode. Here the Ubertooth grabs LAP and UAP to form addresses, and hands-off inquiry to a proper BT dongle. To handle the individual Bluetooth dongles, the hciconfig command was used for bringing up/down of the attached devices, allowing each their turn to work with the Ubertooth One. ubertooth-scan was run with several command line options, the –b option was used to select which attached device to use, while the –t option set a time limit on the duration of the scan (900 for example means 900 seconds, or 15 minutes). The –s option was used to perform a HCI type scan. The –x option turned on the extended query functionality. The testing was run over three phases, with each phase attempting to improve the scripts used.

5.1. Results

A full scan was run and results are shown in Figure 3. The Aircable performed well on the HCI scan, but when compared to the Sena over the three different types of scan, did not perform as well as anticipated for such a powerful device. The laptop device performed well for the Ubertooth scan but was weak elsewhere. Apart from the Linksys modified Ubertooth scan, the two Linksys devices performed poorly, the clear winner during this phase of testing was the Sena Parani UD100. Discoverable versus non-discoverable was again compiled and shown in Figure 4. The proportion of devices with Bluetooth enabled, but discovery mode turned off was again much higher than those devices with Bluetooth and discovery mode both enabled.

The results of runs were merged for discoverability and are displayed in Figure 5. Both non-discoverable and discoverable devices were found during the hcitool type scans only, so this was the criteria used to filter the chart. Clearly shown is a higher number of non-discoverable devices found, particularly by the Aircable and the Sena devices. The Linksys device, despite being recommended in several articles, previously cited, delivered disappointing results. Lastly, the results in Figure 5 were merged for all devices, into Figure 6, to give an idea

of the proportions between what could be found using a regular Bluetooth dongle (discoverable) and using an Ubertooth/Bluetooth-dongle combination (non-discoverable).

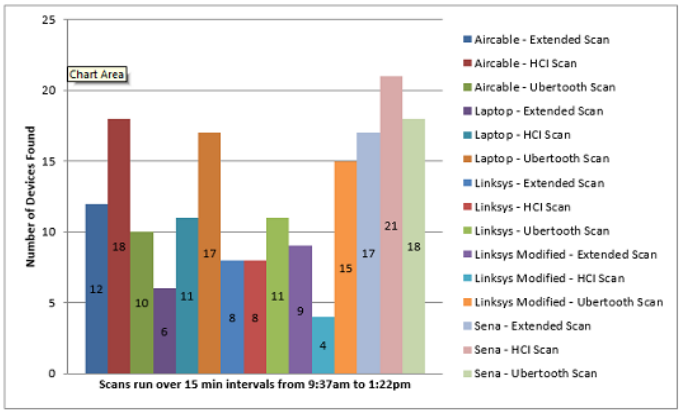

Figure 3. Phase 2 test results

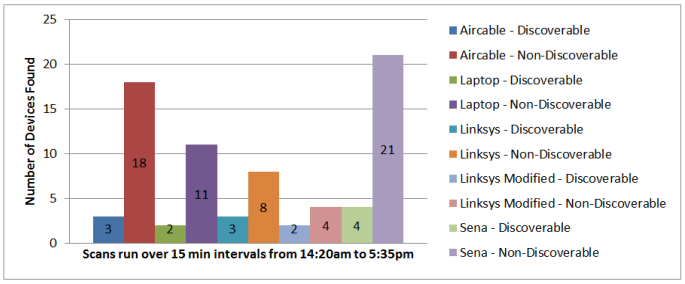

Figure 4. Discoverable versus non-discoverable devices found

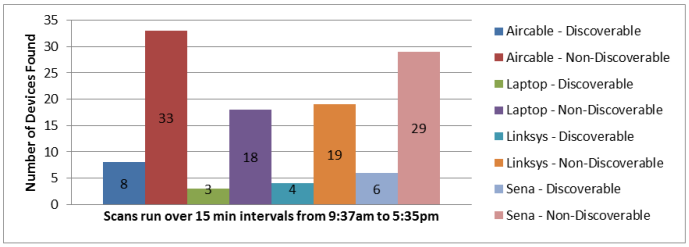

Figure 5. Combined Discoverable versus Non-Discoverable per Device

The graph indicates that, of the Bluetooth-enable devices captured during the testing phase, 17.5% of devices were in discoverable mode, while 82.5% of devices were in non-discoverable mode. While there are several examples of Bluetooth surveys, vulnerable devices [22] or discovering open Bluetooth services [23, 24], they have focussed on the discoverable landscape only. Evidently, auditing Bluetooth devices with a garden variety Bluetooth device alone ignores a significant amount of potentially identifiable devices. While it will not fully complete the picture, the addition of an Ubertooth One can significantly increases the scope of any Bluetooth assessment.

It should be noted that Extended Data Rate (EDR) type devices (Bluetooth v2.0 and above) would not be picked up due to limitations of the Ubertooth hardware, specifically the Texas Instruments Chipcon CC2400 chip used as the radio transceiver interface and the limited to the demodulation capabilities of this chip [25]. The Ubertooth can only capture Bluetooth Basic Rate traffic, it is not able to capture EDR.

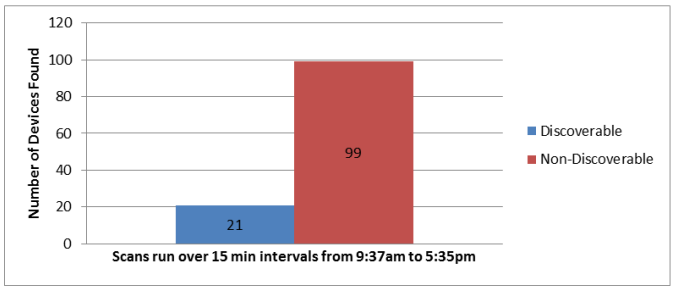

Figure 6. Combined discoverable versus non-discoverable

6. Conclusion

This study showcased how a relatively inexpensive, off-the-shelf commodity device can enumerate multiple Bluetooth devices, even when they are in non-discoverable mode (security advice offered almost universally in the literature). As an inexpensive device for Bluetooth analysis, the Ubertooth is a tremendously valuable tool for security analysts and attackers alike. However, it is also limited in its capabilities to capture Bluetooth Classic network activity. Even though this shows up potential weaknesses in the security landscape, it can also be used to great advantage in the use of logging and auditing of devices, with a view to eventually automating the output and sending SYSLOG type files to a Security Information and Event Management (SIEM) type system for further analysis and investigation.

We show by focusing on Bluetooth Classic devices alone, that there are 4.7 times the amount of Bluetooth devices out there that have Bluetooth enabled but have discovery mode turned off, then those devices that have it turned on. That number underestimates the volume of Bluetooth devices in the wild, as this experiment did not attempt to capture data for EDR devices nor Bluetooth Low Energy devices. This result shows a need for audit features beyond what a regular Bluetooth dongle alone can provide. The Ubertooth One, in combination with the correct Bluetooth hardware can provide the required function of enumerating UAPs and LAPs for potential rogue devices.

Of these devices tested in combination with the Ubertooth One, the Aircable and Sena devices are the most successful, considering their success with capturing data for both discoverable and non-discoverable targets. The Linksys devices were disappointing both in terms of their success at enumerating, but also in terms of the different chipsets used in both devices despite being from the same manufacturer (Broadcom in one case and CSR in the other). The laptop device was weak, but provided a good baseline. Both the Aircable and the Sena devices used the more favoured CSR chipset, plus they had pre-existing RP-SMA connectors. Both would be good choices for a security auditor but the Sena device is 4 times cheaper.

In combination with the right Bluetooth dongle, the Ubertooth One provides a powerful toolset to the compliance auditor's toolbox and can offer invaluable information to any wireless vulnerability assessment. Besides picking out the low hanging fruit of discoverable devices, its ability to identify non-discoverable devices sets it apart, which opens a whole new category of devices that potentially need to be logged, recorded and managed. Limited to pre-EDR versions, this apparent limitation can highlight Bluetooth traffic that runs on older (more vulnerable) versions of the Bluetooth specification, so that mitigation steps can be taken such as organisations migrating BR legacy devices to hardware supporting EDR to mitigate Ubertooth packet capture eavesdropping threats. Using such guidelines as those offered by NIST or attempting to meet the wireless requirements as set out by the Payment Card Industry Data Security Standards, the Ubertooth One offers security auditors a low-cost tool capable of creating asset inventories, while also performing asset discovery, which could be potentially integrated into a SIEM infrastructure, to provide another layer of security to any defence in depth strategy. The number of devices containing Bluetooth chipsets will continue to rise and this area of research will become more and more relevant as security and compliance auditors attempt to stem the tidal wave of vulnerabilities brought by the Bring Your Own Device (BYOD) and Internet of Things (IoT) phenomena.

References

[1] Panse T, Kapoor V, Panse P. A Review on Key Agreement Protocols used in Bluetooth Standard and Security Vulnerabilities in Bluetooth Transmission. *International Journal of Information and Communication Technology Research*. 2012; 2(3): 42-50.

[2] Lee J, Su Y, Shen C. *A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi*. IECON 2007 Industrial Electronics Society Annual Conference of the IEEE, Taipei, Taiwan. 2007: 66-76

[3] Mc Brearty S, Farrelly W, Curran K. The Performance Cost of Preserving Data/Query Privacy Using Searchable Symmetric Encryption. *Security & Communication Networks*. 2016; 9(18): 5311–5332.

[4] Miller B, Bisdikian C. Bluetooth Revealed: The Insider's Guide to an Open Specification for Global Wireless Communication. 2001. ISBN; 0-13-090294-2, Prentice Hall PTR, NJ, USA.

[5] Chien H. SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity. IEEE Transactions on Dependable and Secure Computing. 2007; 4(4): 337-340.

[6] Ossmann M. Project Ubertooth: Discovering the Bluetooth UAP. 2014. http://ubertooth.blogspot.ie/2014/06/discovering-bluetooth-uap.html

[7] Ryan M. BLE Fun with Ubertooth: Sniffing Bluetooth Smart and Cracking Its Crypto. 2014. [Online] http://blog.lacklustre.net/posts/BLE_Fun_With_Ubertooth:_Sniffing_Bluetooth_Smart_and_Cracking_Its_Crypto/

[8] Padgette J, Scarfone K, Chen L. Guide to Bluetooth Security, s.l.: U.S. Department of Commerce. NIST Special Publication. 2012. 800-121 Revision 2. Available at: http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-121r2.pdf

[9] Haataja K, Toivanen P. Two practical man-in-the-middle attacks on Bluetooth secure simple pairing and countermeasures. *IEEE Transactions on Wireless communications*. 2010; 9(1): 22-34.

[10] Tipton HF. Official (ISC)2 Guide to the CISSP CBK. 2nd ed. s.l.: CRC Press. 2010.

[11] Snow C, Primak S. *Performance evaluation of TCP/IP in Bluetooth based systems. Vehicular Technology Conference.* IEEE 55th Vehicular Technology Conference. VTC Spring 2002 (Cat. No.02CH37367). 2002; 1(1): 429-433

[12] Vance P, Prasad G, Harkin J, Curran K. Designing a Compact Wireless Network based Device-free Passive Localisation System for Indoor Environments. *International Journal of Wireless Networks and Broadband Technologies (IJWNBT)*. 2015; 4(2): 28-43.

[13] Wright J, Cache J. Hacking Exposed Wireless. 3rd ed. New York: McGraw Hill Education. 2015.

[14] Cope P, Campbell J, Hayajneh T. *An investigation of Bluetooth security vulnerabilities*. IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV. 2017: 1-7.

[15] Cisco 2014. Wireless Networking and PCI Compliance Networking Solutions White Paper, San Jose: CISCO.Available at: https://www.cisco.com/c/dam/en/us/solutions/collateral/enterprise-networks/pci-compliance/at_a_glance_c45-639503.pdf

[16] Korba AA, Nafaa M, Ghanemi S. Hybrid Intrusion Detection Framework for Ad hoc networks. *International Journal of Information Security and Privacy (IJISP).* 2016; 10(4): 1-32.

[17] Souppaya M, Scarfone K. Guidelines for Managing the Security of Mobile Devices in the Enterprise, s.l.: U.S. Department of Commerce. Special Publication. 2013. (NIST SP)-800-124 Rev 1. Available at: https://www.nist.gov/publications/guidelines-managing-security-mobile-devices-enterprise

[18] Scarfone K, Mell P. Guide to Intrusion Detection and Preventions Systems (IDPS), s.l: U.S. Department of Commerce. 2007.

[19] Nardi T. Bluetooth Hunter's Guide. 2600: The Hacker Quarterly. 2012; 29 (2) ASIN: B004GB1WF6.

[20] Huang AS, Rudolph L. Bluetooth Essentials for Programmers. New York: Cambridge. 2007.

[21] Kumawat A, Sharma AK, Kumawat S. Identification of Cryptographic Vulnerability and Malware Detection in Android. *International Journal of Information Security and Privacy (IJISP)*. 2017; 11(3): 15-28.

[22] Holeman R. Passive Aggressive Bluetooth Scanning with Python. 2013. [Online] Available at: http://www.hackgnar.com/2014/03/passive-aggressive-bluetooth-scanning.html

[23] Talal B, Rachid M. Service Discovery-A Survey and Comparison. arXiv:1308.2912. 2013. Online at: http://adsabs.harvard.edu/abs/2013arXiv1308.2912T

[24] Curran K, Dempsey S. Enhancing Bluetooth Security for M-Commerce Transactions. *Advanced Engineering in Informatics Journal.* 2011; 36(1): 3519-3523.

[25] Cullen G, Curran K, Santos J, Toman M. *Using Cooperatively Applied Positioning Techniques Utilizing Range Extension to Manage Passengers in Crowded Environments*. 2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN), France. 2018: 1-10.